# Optimal Latency Guarantee for Multiple Concurrent Packets with New IP

**Presenter: Lijun Dong**
**Principal Architect**

Authors: Lijun Dong, Richard Li
Network Technology Lab
Futurewei Technologies Inc.

Contact: lijun.dong@futurewei.com

IARIA ACCSE 2022

FUTUREWEI Technologies

# New IP and Fundamental Components

| IP Header | Contract | User Payload |
|---|---|---|

**A packet carries a contract between an application and the network.**
**The network and routers fulfill the contract.**

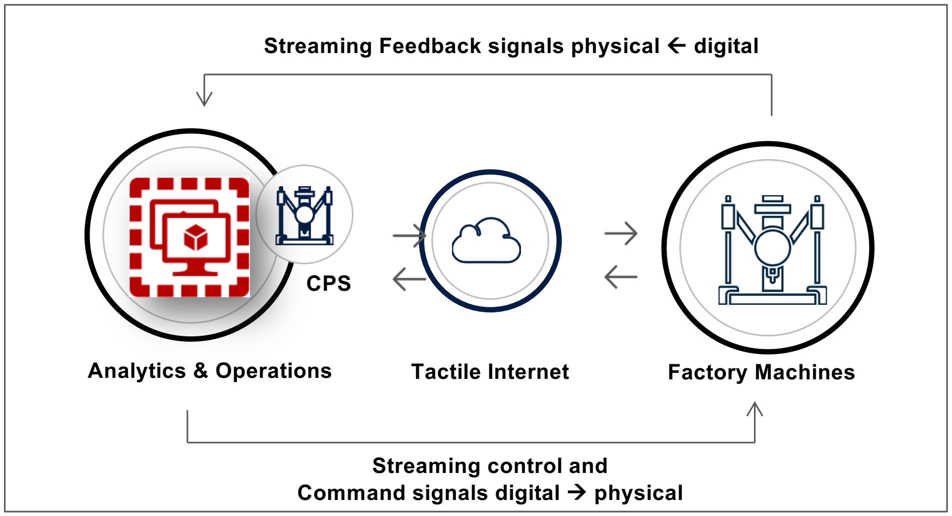
**<Contract>** := <Contract Clause> | <Contract Clause> AND <Contract>

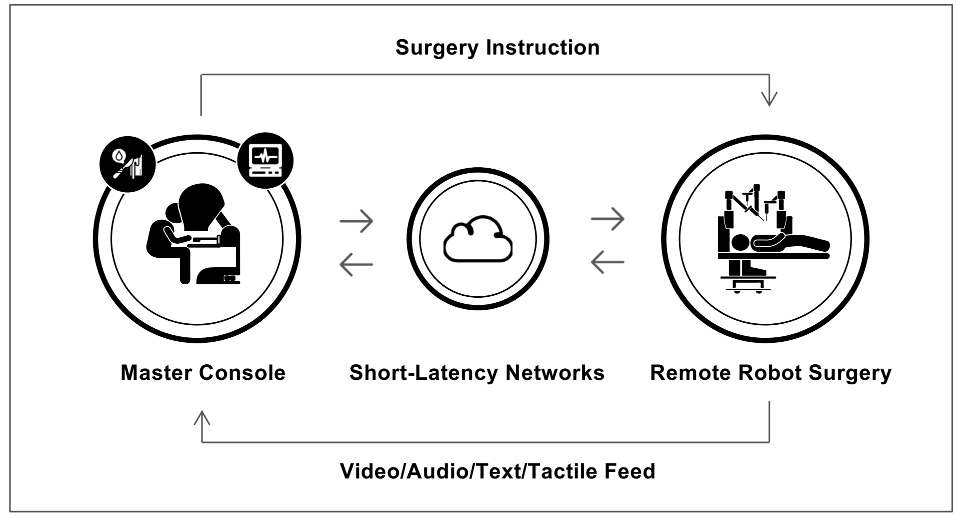
**<Contract Clause>** := <Contract Item> | <Contract Item> OR <Contract Clause>

**<Contract Item>** := <Event, Condition, Action> | <Meta Data> | <Event, Condition, Action> <Meta Data>

**<Event, Condition, Action>** := <Action> | <Condition> <Action> | <Event> <Condition> <Action>

# Tactile Internet and Short-Latency Networks




- **Ultra-low latency**
  Sub-millisecond to 5 milliseconds.
- **Ultra-low loss**
  Loss of packets is almost intolerable
- **Ultra-high bandwidth**
  From 360-degree video to holograms. VR feed: 5 Gbps; Holograms: Tbps
- **Stringent synchronization**
  Different human-brain reaction times to different sensory inputs (tactile: 1ms, visual: 10ms, or audio: 100ms). Hence real-time feedback from different inputs must be synchronized accordingly.
- **Differentiated prioritization levels**
  Prioritizing streams based on their immediate relevance.

# Assumptions for Single Flow Scenario

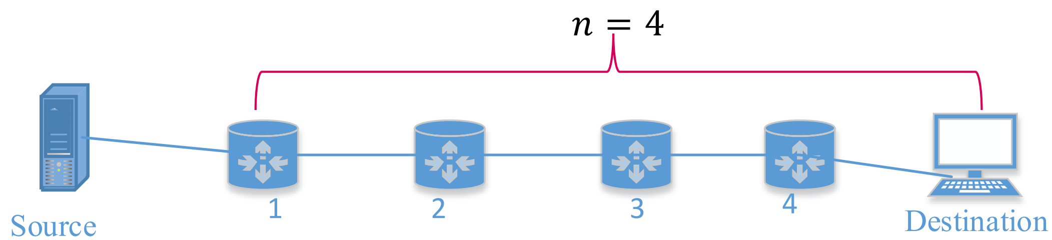
- For a particular flow, we consider the scenario, in which the network will need to guarantee the in-time delivery of the packets belonging to that flow, regardless of other concurrent flows. In other words, this particular flow has the highest priority when being scheduled to be forwarded.

- We consider the end-to-end in-time delivery requirement is set as:
  - $latency \leq d$



FUTUREWEI Technologies

# Assumptions




- The first router that the packet from the source reaches is able to know the number of hops between itself and the destination, which is denoted as $n$.
- There are $n$ number of hops between the first router and the destination, which means $n$ number of routers involved in the packet forwarding.
- The source can specify the in-time delivery requirement in the Contract Clause of the New IP packet: $latency \leq d$
- The metadata of the New IP packet can carry how much time has passed since the packet is sent out from the source. It is initialized to 0 by the source.

# New IP Packet Construction

| Contract Clause | Metadata | | |
|---|---|---|---|
| $latency \leq d$ | time passed ($t$) | allowance for the current router ($budget$) | Number of hops away from the destination ($m$) |

- When the packet reaches the router 1, $t$ is set to the time used to transport the packet from the source to the router 1, $m$ is set to $n$.
- The current router is allowed to have the time budget to forward the packet to the next router. The time budget has two aspects:
  - If the router uses less time than the budget, then it does not affect the following routers, but gives them more time budget to use.
  - If the router uses more time than the budget, it will affect the rest of the routers. However, it does not mean the packet has to be dropped if the budget cannot be met. The hybrid policy used at the router for the particular flow is that it puts the packet at the highest priority, but tries its best.
- The allowance for the current router is calculated as
  - $budget = \frac{d-t}{m}$

# New IP Packet Processing

- When the packet reaches the intermediate routers (e.g., the router 2, 3 and 4) , $t$ is set to the time used to transport the packet from the source to the current router, $m$ is deducted by 1 every time the packet is being forwarded by a router, *budget* is calculated accordingly.
- When an intermediate router finds that the residual time ($d$-$t$) is still not enough for it to transfer the packet to the destination, the packet is dropped and the in-time guarantee fails because it is not a realistic requirement.
- The intermediate router needs to reply the source with Metadata indicating the failure as well as the suggested in-time latency upper bound in the Command block for the source to reference to.

| Contract Clause | Metadata | |
|---|---|---|
| $latency \leq d$suggested | unsuccessful | Number of hops away from the source ($n$-$m$) |

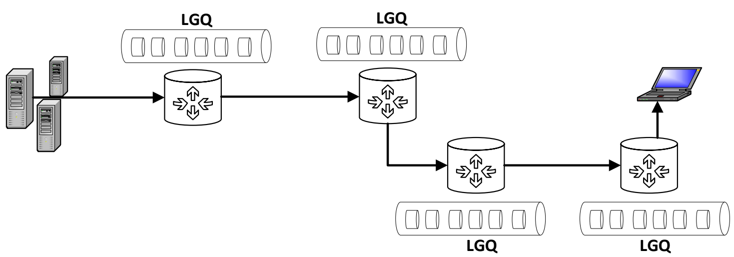


FUTUREWEI Technologies

# Multiple Flows Scenario

- In realistic, routers might need to guarantee the in-time delivery for packets from multiple flows.

| Packet number | Budget | Time duration that the packet stays in the router |
|---|---|---|
| p1 | $b_1$ | $P_{\sigma(1)}$ |
| p2 | $b_2$ | $P_{\sigma(2)}$ |
| p3 | $b_3$ | $P_{\sigma(3)}$ |




- The time duration that a packet stays in the router depends on how the other packets are scheduled. A packet's time duration equals to the time duration of the packets scheduled before it and its own header processing, propagation and transmission delay at the router.



FUTUREWEI Technologies

# Optimization Problem Formulation

- The optimization problem is formulated with the objective to minimize the total amount of time generated by the current router:

$$\min \sum_{k=1}^{K} t_k$$

$$s.j.: t_k \leq b_k \text{ for } k = 1 \ldots K$$

- The problem is that we want to find a permutation of $\{1, \ldots K\}$ $(\sigma: \{1, \ldots K\} \rightarrow \{1, \ldots K\})$ which represents the scheduling of the packet forwarding in the queue (i.e., the position in the queue), such that the total waiting time can be minimized.



FUTUREWEI Technologies

# Optimization Problem Formulation

- It can be calculated that:
  - $t_{\sigma(k)} = \sum_{i=1}^{k} P_{\sigma(i)}$, where $P_{\sigma(i)}$ is the processing time when it is served.
- The optimization problem is converted to:

$$\min \sum_{k=1}^{K} t_{\sigma(k)}$$

$$s.j.: t_k \leq b_k \text{ for } k = 1 \ldots K$$

- $\sum_{k=1}^{K} t_{\sigma(k)} = K * P_{\sigma(1)} + (K-1) * P_{\sigma(2)} + \ldots + P_{\sigma(K)}$

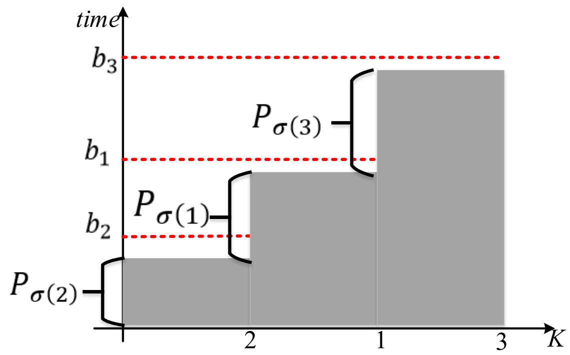


FUTUREWEI Technologies

# Simple Example

- Simple Example: $\sigma: \{1,2,3\} \rightarrow \{2,1,3\})$
  - $t_1 + t_2 + t_3 = P_{\sigma(2)} + \left(P_{\sigma(2)} + P_{\sigma(1)}\right) + \left(P_{\sigma(2)} + P_{\sigma(1)} + P_{\sigma(3)}\right)$
    $= 3 * P_{\sigma(2)} + 2 * P_{\sigma(1)} + P_{\sigma(3)}$

- The problem is firstly to find all feasible solutions, then find the optimal one that minimizes the area in grey.



# Backtracking Algorithm

| $backtracking(objective, constraint, s)$ |
|---|
| if $discard(constraint, s)$, then |
| return; |
| if $accept(constraint, s)$, then |
| $record(constraint, s)$ |
| $o = calculate(objective, s)$ |
| $select(smallest, objective, s)$ |
| $\text{a} = first(constraint, s)$ |
| while $\text{a} \neq \text{NULL}$ do |
| $backtracking(objective, constraint, a)$ |
| $a = next(constraint, a)$ |

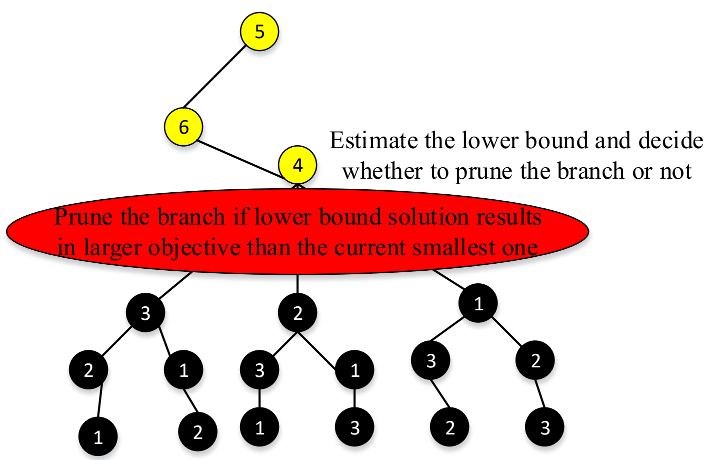
- $discard(constraint, s)$: return true only if the partial scheduling $s$ is not worth going further.
- $accept(constraint, s)$: return true if $s$ is a solution that satisfies all constraints, and false otherwise.
- $first(constraint, s)$: generate the first extension of candidate $s$, which means the first packet in the queue is selected and added to $s$.
- $next(constraint, a)$: generate the next alternative extension of a candidate, after the extension $a$, which means another different packet is selected to be next in the queue.
- $record(constraint, s)$: record the solution that satisfies all constraints.
- $calculate(objective, s)$: calculate the objective result for the solution.
- $select(smallest, objective, s)$: compare the current solution with the selected solution to make sure the selected solution always has the minimal objective result.



FUTUREWEI Technologies

# Backtracking Algorithm with Branch and Bound (BABB)

- Enumeration procedure can find the optimal solution for any problem with constraints. But it takes too much time, even with the proposed algorithm using backtracking, if the number of packets that needs to be scheduled is large at a router.
- So we improve the algorithm by leveraging Branch and Bound concept:
  - For a current extension that is partial, estimate the lower bound, stop extending from the current candidate and prune the whole branch rooting from it.




- The lower bound can easily be calculated by sorting the budgets in decreasing order. For example, for packet p1, p2, p3, the budgets b1, b2, b3 in decreasing order are b2, b1, b3, thus the branch of 2, 1, 3 results in the lower bound of the entire branch extended from 5, 6, 4.
- If this lower bound solution (i.e., 5, 6, 4, 2, 1, 3) cannot obtain a smaller objective than the current smallest one, then the entire branch should not be traversed and pruned from the recursive iteration.

# Accelerate the Algorithm by Stopping at One/First Feasible Solution (OPFS)

- In order to significantly reduce the running time of the algorithm, instead of minimizing the objective, the algorithm can be stopped when the first solution that satisfies the constraints is found.

- Select the packet with the minimal processing time as the first packet in the solution, then extend the solution with the packet with the second minimal processing time as the second packet, and so on. With this improvement, it is more likely to get a solution that satisfies the constraints.

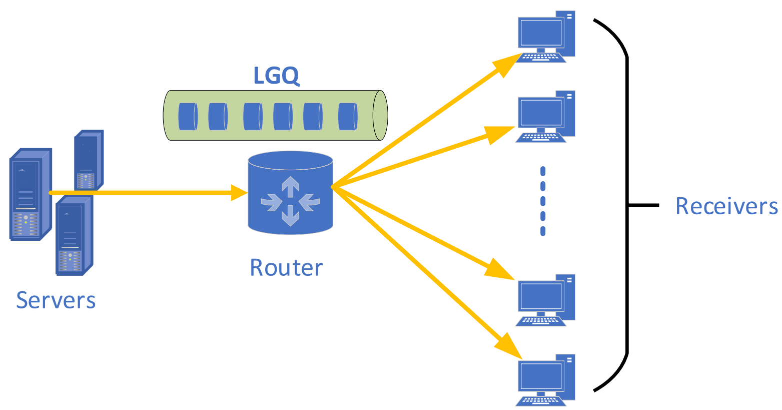
# Performance Evaluation

- Two types of performances are being evaluated: (1) packet delivery success ratio, which is defined the ratio of the packets that get scheduled appropriately and meet their corresponding deadlines. (2) Average stay time improvement. We compare the performances of the proposed BABB, OPFS algorithms with:

    - First In First Out (FIFO): The packets are scheduled according to their arrival time at the outgoing queue of the router. The packet which arrives earliest is scheduled firstly.

    - Smallest Transmission Time First (STTF): the packets are scheduled based on the incremental order of the transmission time. The transmission time is proportional to the packet size if the outgoing link bandwidth is fixed. Thus, in STTF, the minimum-sized packet is scheduled firstly.

    - Largest Transmission Time First (LTTF): the packets are scheduled based on the decremental order of the transmission time. Thus, in LTTF, the bulkiest packet is scheduled firstly.

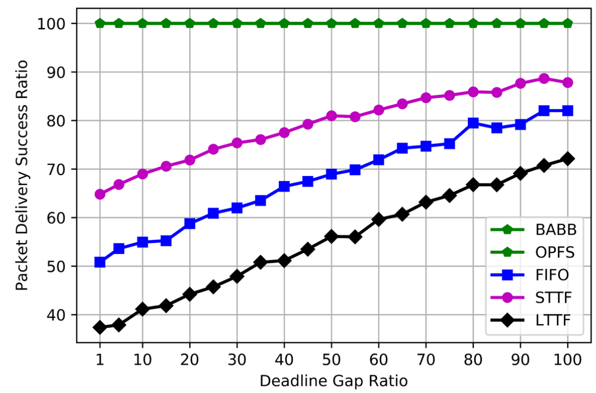


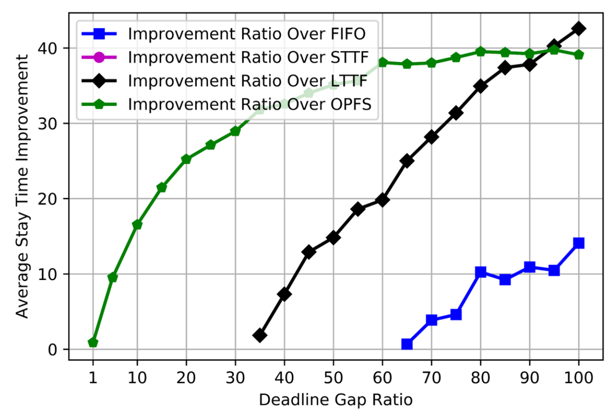



FUTUREWEI Technologies

# Impact of Deadline Gap Ratio

- We specifically design the transmission time and deadline for the packets in LGQ: firstly we random generate the transmission time of the packets in the range [1,10] ms, then the deadline of each packet is assigned by adding some extra time compared to the dwell time.
- Deadline gap ratio is defined as the ratio between the upper bound of this additional time and the transmission time upper bound. For example, if the deadline gap is 2, then the deadline of a packet is given by adding a random number between [0, 10*2] = [0, 20] ms to the dwell time.

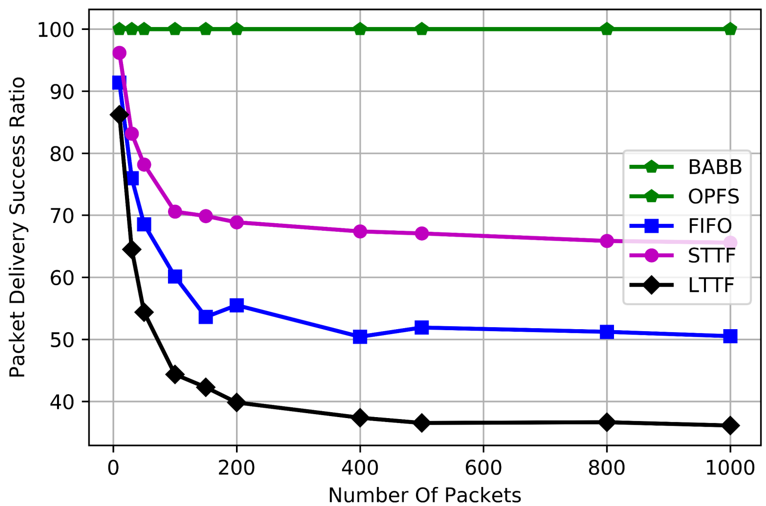


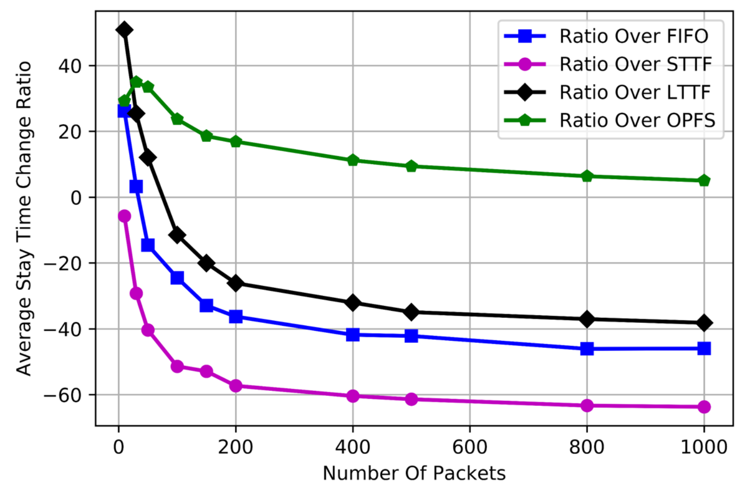






FUTUREWEI Technologies

# Impact of Number of Latency-Sensitive Packets in LGQ



# Conclusions

- The paper leverages the New IP framework to carry in-time guarantee contract, as well as associated deadline constraint and other metadata in the packet, such that each intermediate router on the path from the source to the destination can execute more sophisticated scheduling on multiple packets on the same outgoing port instead of traditional statistical multiplexing.

- The proposed backtracking solution with bound-and-branch improvement can achieve the minimal average stay time of the packets which require in-time guarantee, and the successful delivery ratio of packets is maximized compared to any other scheduling schemes.

# Thank You!

Contact: lijun.dong@futurewei.com